



Haga lo que hicimos en el aula: Creando la excepción

En el proyecto *bytebank-heredado-cuenta*, vamos a refactorizar el código de la clase *Cuenta* y crearemos una excepción.

1) En primer lugar, cree nuestra excepción *SaldoInsuficienteException*:

```
public class SaldoInsuficienteException extends Exception{  
  
    public SaldoInsuficienteException(String msg) {  
        super(msg);  
    }  
}
```

[COPIA EL CÓDIGO](#)

2) Ahora, abra la clase *Cuenta*. Busque el método *sacar()* y cambie el tipo de retorno de *boolean* a *void*. Elimina los* *returns** y lanza la nueva excepción. El método *sacar()* se verá así:

```
public void sacar(double valor) throws SaldoInsuficienteExcept:  
  
    if(this.saldo < valor) {  
        throw new SaldoInsuficienteException("Saldo: "  
    }  
  
    this.saldo -= valor;  
}
```

[COPIA EL CÓDIGO](#)

3) Tenga en cuenta que hemos invertido la lógica para poder lanzar la excepción primero.

4) Vea que ahora nuestro método de *transferir()* también necesita ser cambiado, ya que ahora ya no tendrá un retorno del tipo *boolean* y el método *sacar()* no devuelve *void*, además de la excepción en la firma del método.

5) Cambie el método para que parezca como:

```
public void transferir(double valor, Cuenta destino) throws Sa:  
    this.saca(valor);  
    destino.depositar(valor);  
}
```

[COPIA EL CÓDIGO](#)

6) Debido a esto, necesitaremos cambiar nuestro método *sacar()* de nuestra clase *Cuenta Corriente*:

```
@Override  
public void sacar(double valor) throws SaldoInsuficienteExcept:  
    double valorASacar = valor + 0.2;  
    super.saca(valorASacar);  
}
```

[COPIA EL CÓDIGO](#)

7) Ahora, cambie la clase *TestCuenta* para que funcione con nuestras excepciones. Para hacer esto, agregue un "throws" a la firma del método *main*:

```
public class TestCuenta {
```

```
public static void main(String[] args) throws SaldoInsuficienteException {  
  
    CuentaCorriente cc = new CuentaCorriente (111, 111);  
    cc.depositar(100.0);  
  
    CuentaAhorro ca = new CuentaAhorro(222, 222);  
    ca.depositar(200.0);  
  
    cc.transfiere(110.0, ca);  
  
    System.out.println("CC: " + cc.getSaldo());  
    System.out.println("CA: " + ca.getSaldo());  
}  
}
```

[COPIA EL CÓDIGO](#)

8) Luego, intente transferir un valor no válido y ejecute el código:

9) Finalmente, cree una clase *TestSacar* para probar el método *sacar*. Utilice un *try-catch* para capturar la excepción:

```
public class TestSacar {  
  
    public static void main(String[] args) {  
        Cuenta cuenta = new CuentaCorriente(123, 321);  
  
        cuenta.depositar(200.0);  
  
        try {  
            cuenta.sacar(210.0);  
        } catch (SaldoInsuficienteException ex) {  
            System.out.println("Exception: " + ex.getMessage());  
        }  
    }  
}
```

```
        ex.printStackTrace();  
    }  
  
    System.out.println(cuenta.getSaldo());  
}  
}
```

[COPIA EL CÓDIGO](#)

10) ¡Ejecute la clase *TestSacar*!